WHAT IS CLAIMED IS:

1. A context controller for managing multitasking in a processor, comprising:

a time slice instruction counter that counts a number of instructions executed with respect to a given background task; and

a background task controller that cyclicly activates a context corresponding to another background task when said number equals a dynamically-programmable time slice value.

2. The context controller as recited in Claim 1 wherein said time slice instruction counter initially contains said dynamically-programmable time slice value as a time slice for said given background task begins, said time slice instruction counter decrementing as said instructions with respect to said given background task are executed.

3. The context controller as recited in Claim 1 wherein said context controller places said processor in an idle state when all of said background tasks are inactive.

4. The context controller as recited in Claim 1 wherein said background task controller is adapted to activate a context

corresponding to a particular background task by vectoring to a software-selectable memory location.

5. The context controller as recited in Claim 1 further comprising a foreground task controller that activates contexts corresponding to foreground tasks based on priority and in response to events, said background task controller cyclicly activating contexts corresponding to said background tasks subject to activation of said contexts corresponding to said foreground tasks.

6. The context controller as recited in Claim 1 wherein said dynamically-programmable time slice value is contained in a register of said processor.

7. The context controller as recited in Claim 1 wherein application tasks executing on said processor can program said dynamically-programmable time slice value.

8. A method of managing multitasking in a processor, comprising the steps of:

counting a number of instructions executed with respect to a given background task; and

cyclicly activating a context corresponding to another background task when said number equals a dynamically-programmable time slice value.

9. The method as recited in Claim 8 wherein said step of counting comprises the steps of:

initializing a time slice instruction counter with said dynamically-programmable time slice value as a time slice for said given background task begins; and

decrementing said time slice instruction counter as said instructions with respect to said given background task are executed.

10. The method as recited in Claim 8 further comprising the step of placing said processor in an idle state when all of said background tasks are inactive.

11. The method as recited in Claim 8 wherein said step of cyclically activating comprises the step of vectoring to a software-selectable memory location.

12. The method as recited in Claim 8 further comprising the step of activating contexts corresponding to foreground tasks based on priority and in response to events, said step of cyclically activating comprising the step of cyclicly activating contexts corresponding to said background tasks subject to activation of said contexts corresponding to said foreground tasks.

13. The method as recited in Claim 8 further comprising the step of storing said dynamically-programmable time slice value in a register of said processor.

14. The method as recited in Claim 8 further comprising the step of programing said dynamically-programmable time slice value with application tasks executing on said processor.

15. A processor, comprising:

an instruction decoder that decodes instructions received into said processor and corresponding to a plurality of tasks;

a plurality of register sets, corresponding to said plurality of tasks, that contain operands to be manipulated;

an execution core, coupled to said instruction decoder and said plurality of register sets, that executes instructions corresponding to an active one of said plurality of tasks to manipulate ones of said operands; and

a context controller, coupled to said instruction decoder and said execution core, that manages multitasking with respect to said plurality of tasks, including:

a time slice instruction counter that counts a number of instructions executed with respect to a given background task, and

a background task controller that cyclicly activates a context corresponding to another background task when said number equals a dynamically-programmable time slice value.

16. The processor as recited in Claim 15 wherein said time slice instruction counter initially contains said dynamically-programmable time slice value as a time slice for said given background task begins, said time slice instruction counter decrementing as said instructions with respect to said given background task are executed.

17. The processor as recited in Claim 15 wherein said context controller places said processor in an idle state when all of said background tasks are inactive.

18. The processor as recited in Claim 15 wherein said background task controller is adapted to activate a context corresponding to a particular background task by vectoring to a software-selectable memory location.

19. The processor as recited in Claim 15 wherein said context controller further includes a foreground task controller that activates contexts corresponding to foreground tasks based on priority and in response to events, said background task controller cyclicly activating contexts corresponding to said background tasks subject to activation of said contexts corresponding to said foreground tasks.

20. The processor as recited in Claim 15 wherein said dynamically-programmable time slice value is contained in a register of said processor.

21. The processor as recited in Claim 15 wherein application tasks executing on said processor can program said dynamically-programmable time slice value.

22. The processor as recited in Claim 15 wherein said processor forms a portion of a general-purpose computer.